

A Route Confidence Evaluation Method for Reliable Hierarchical Text Categorization

Nima Hatami¹, Camelia Chira², and Giuliano Armano³

¹ BioCircuits Institute

University of California

San Diego, CA 92093-0328, USA

² Department of Computer Science

Babes-Bolyai University

Kogalniceanu 1, Cluj-Napoca 400084, Romania

³ Department of Electrical and Electronic Engineering

University of Cagliari

Piazza D'Armi, I-09123 Cagliari, Italy

Abstract. Hierarchical Text Categorization (HTC) is becoming increasingly important with the rapidly growing amount of text data available in the World Wide Web. Among the different strategies proposed to cope with HTC, the Local Classifier per Node (LCN) approach attains good performance by mirroring the underlying class hierarchy while enforcing a top-down strategy in the testing step. However, the problem of embedding hierarchical information (parent-child relationship) to improve the performance of HTC systems still remains open. A confidence evaluation method for a selected route in the hierarchy is proposed to evaluate the reliability of the final candidate labels in an HTC system. In order to take into account the information embedded in the hierarchy, weight factors are used to take into account the importance of each level. An acceptance/rejection strategy in the top-down decision making process is proposed, which improves the overall categorization accuracy by rejecting a few percentage of samples, i.e., those with low reliability score. Experimental results on the Reuters benchmark dataset (RCV1-v2) confirm the effectiveness of the proposed method, compared to other state-of-the-art HTC methods.

1 Introduction

Text categorization is one of the key tasks in information retrieval and text mining. It is widely used in many intelligent systems, e.g., content-based spam filtering, e-mail categorization, web page classification and digital libraries [3] [4] [5]. Due to some challenging characteristics, such as the huge number of sparse features and a typically large number of classes, text categorization attracted a lot of attention from different research fields, including machine learning, data mining and pattern recognition.

There are three main approaches to text categorization: (i) flat approaches, which totally ignore the class hierarchy, (ii) local approaches,

which run a classifier only for a subset of the hierarchy, and (iii) big-bang approaches, which use a single classifier for the whole category space. However, despite the variety of proposed methods, also depending on different types of classifiers and on feature selection/extraction algorithms, there is no clear outperforming method (see [6] and [7] for more information on this issue).

The main idea of Hierarchical Text Categorization (HTC) is to take benefit of the information embedded in the hierarchical structure, with the goal of improving the classification performance. Browsing the massive amount of data represents a further motivation for using a hierarchical structure. Typically, categories are structured according to a top-down view, where nodes at upper level are used to represent generic concepts while nodes at lower levels are viewed as more specific categories. Top-down error propagation is a major disadvantage of HTC methods, which implies that a misclassification made at upper levels cannot be recovered at lower levels. Some *error correction* strategies have been proposed to minimize error propagation [7], but their performance is still limited.

According to the survey paper of Silla and Freitas [6], there are three kinds of local classifier methods, depending on how local information is used and on how local classifiers are built: i) local classifier per node (LCN), ii) local classifier per parent node (LCP), and iii) local classifier per level (LCL). Each of these approaches has its own drawbacks and benefits. In the first, each node and its corresponding classifier is independent from the rest of the hierarchy, thus facilitating the maintenance of the hierarchy, as (to some extent) the classifier associated to a node can be modified without manipulating the others. While LCN methods employ a great number of classifiers, one for each node of the given hierarchy, LCP and LCL methods lie on non binary classifiers –which is a clear source of additional complexity for the underlying learning process. In this paper, an evaluation strategy for LCN methods is proposed,⁴ which allows to evaluate the route of the underlying hierarchy that has been selected depending on the input in hand. The evaluation strategy returns a reliability measure, with the goal of deciding the confidence of the final label assignment. Weight factors for each level of the hierarchy are used, with the goal of adding hierarchical information in the decision making process. Each weight factor is strictly related with the likelihood for an error to occur at the given level. A thresholding mechanism is proposed to accept/reject the candidate label by considering the reliability score assigned to the candidate route in the hierarchy. Experimental results show a significant increase in categorization accuracy, obtained by rejecting a few percentage of the samples with low reliability score.

The remainder of this paper is divided as follows: Section 2 briefly recalls the LCN approach and its training/testing strategies. The proposed route reliability evaluation is described in Section 3. In Section 4, we validate our proposed method on three topics, industry and regions datasets of the RCV1-v2, the Reuter’s text categorization test collection and discuss the results. Section 5 concludes the paper.

⁴ It is worth pointing out that, although fraed for LCN, the strategy could be easily adapted to any other top-down local classifier methods.

2 The Local Classifier per Node Approach

LCN appears to be the most used and acknowledged approach in the hierarchical classification literature [6]. A local binary classifier runs on each node of a hierarchy except for the root node (whose typical responsibility is to dispatch the input to be classified to all its children). The hierarchical information and parent-child relationship is taken into account by defining the set of positive and negative examples while training each classifier. The decision making process starts from the root node and proceeds downward to the lower levels of a hierarchy. Figure 1 illustrates this approach with an example.

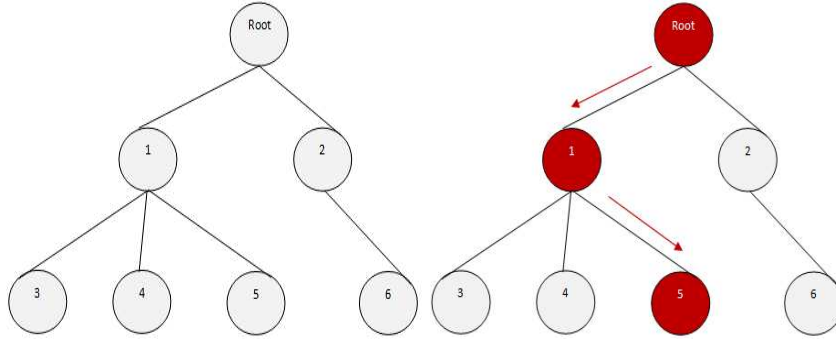


Fig. 1. Some relevant features of LCN methods: a case of inconsistency (on the left) and a typical top-down decision strategy in action (on the right).

Being \mathcal{T} the training set, A_i examples whose most specific class is c_i , $T_{c_i}^+$ and $T_{c_i}^-$ positive and negative training set of c_i , there are many training policies as follows:

- *Exclusive policy* [8]: $T_{c_i}^+ = A_i$ and $T_{c_i}^- = T - T_{c_i}^+$
- *Less exclusive policy* [8]: $T_{c_i}^+ = A_i$ and $T_{c_i}^- = T - (A_i \cup \Downarrow A_i)$ where $\Downarrow A_i$ is the set of descendent categories of A_i .
- *Less inclusive policy* [8]: $T_{c_i}^+ = A_i \cup \Downarrow A_i$ and $T_{c_i}^- = T - T_{c_i}^+$
- *Inclusive policy* [8]: $T_{c_i}^+ = A_i \cup \Downarrow A_i$ and $T_{c_i}^- = T - (A_i \cup \Downarrow A_i \cup \Uparrow A_i)$ where $\Uparrow A_i$ is the set of ancestor categories of A_i .
- *Siblings policy* [9]: $T_{c_i}^+ = A_i \cup \Downarrow A_i$ and $T_{c_i}^- = \Leftrightarrow c_i \cup \Downarrow (\Leftrightarrow c_i)$ where $\Leftrightarrow c_i$ is the set of sibling categories of A_i .
- *Exclusive siblings policy* [10]: $T_{c_i}^+ = A_i$ and $T_{c_i}^- = \Leftrightarrow c_i$

The testing step can be performed in several ways. In the event that the output of each classifier is separately calculated for any incoming sample,

this decision strategy is naturally multi-labeled. On the other hand, class-membership inconsistency may occur. To show a case of inconsistency, let us consider a case in which a sample belongs to nodes 1, 5, 2, and 6, while in fact the classifier that corresponds to node 2 has not been fired –see Figure 1 (left). This event, not so unlikely to occur, shows that some LCN methods are prone to class-membership inconsistency. Some methods have been devised to avoid inconsistencies, which force the selection of only one node at each level of the hierarchy [11] [12] [13]. The top-down strategy is a commonly-used approach in LCN methods to avoid inconsistencies. This strategy assumes that the evaluation starts from the root and goes downward to the leaf –as shown in Figure 1 (right). At each level of the hierarchy, except for the root, the decision about which node to select at the current level is also based on the node predicted at the previous (parent) level. For example, suppose that the output of the local classifier for class 1 is true, and the output of the local classifier for class 2 is false. At the next level, the system will only consider the output of classifiers predicting classes which are children of class 1, i.e., nodes 3, 4 and 5.

Any top-down approach in which a stopping criterion permits the classification process to stop at any internal node of the underlying hierarchy is prone to the so-called “blocking problem”, which occurs when a classifier at a certain level in the class hierarchy predicts that the sample does not have the class associated with that classifier. In this case the sample will be “blocked”, i.e., it will not be passed to the descendants of that node. This phenomenon happens whenever a threshold is used at each node, and if the confidence score or posterior probability of the classifier at a given node (for a given test sample) is lower than this threshold, the classification disregards the incoming sample.

Moreover, top-down methods were originally forced to predict a leaf node, also known as mandatory leaf-node prediction in the literature. It is worth pointing out that a non mandatory leaf-node prediction setting, in combination with a top-down approach, does not prevent the blocking problem to occur, as the process can be stopped also due to an erroneous classification (false negative).

3 The Proposed Label Evaluation Method for LCN

In this section, we present the proposed label evaluation method for the LCN approach which enforces a top-down strategy for the testing phase. The proposed method tries to ensure the reliability of a candidate route in the hierarchy for a test sample before assigning the final label by the classifier. The idea is to identify the samples likely to be assigned the “false” label while they are in fact true. Once this is achieved, there are two options: to send the sample to another classification process or to simply reject the sample and send it to the manual labeling process. This decision is particularly crucial for the applications associated with a high cost of mislabeling true positives.

In the proposed method, we calculate the *confidence score* for each selected node at each level of hierarchy as follows:

$$CS(\hat{c}) = \frac{\mathcal{P}(\hat{c})}{\sum_{\hat{c} \leftrightarrow \hat{c}} \mathcal{P}(\hat{c} \leftrightarrow \hat{c})} \quad (1)$$

where \hat{c} is a node and $\mathcal{P}(c)$ is its posterior probability. This measure takes into account the confidence of the selected node compared to the rest of its siblings.

Furthermore, in order to include the hierarchical information embedded in parent-child class relationships, weight factors are computed for each level of the hierarchy. These weights are calculated based on the accuracy of each level, so that a level with high error rate gets a reduced weight factor. While performing top-down evaluation of a sample, we calculate the *reliability score* for the candidate route using formula 2.

Finally, using a threshold to decide about the label assigned to the candidate route generates an accept/reject answer or the application of another classifier designed for this purpose.

The threshold determined by Equal Error Rate (EER) leads to the equal false acceptance (FA) and false rejection (FR) rates. However, other strategies can also be considered. The proposed method is sketched in Algorithm 1 with more detail.

4 Experimental Results

Dataset description

The Reuters Corpus Volume I (RCV1) [2] is a benchmark dataset widely used in text categorization and in document retrieval. It consists of over 800,000 newswire stories, collected by the Reuters news and information agency. The stories have been manually coded using three orthogonal category sets. Category codes from three sets (Topics, Industries, and Regions) are assigned to stories:

- Topic codes capture the major subject of a story. The hierarchy of topics consists of a set of 104 categories organized in a four-level hierarchy.
- Industry codes are assigned on the basis of the types of business discussed in the story.
- Region codes include both geographic locations and economic/political groupings.

We pre-processed documents proposed by Lewis et al. by retaining only documents associated to a single category. This choice depends on the fact that in this study we are interested in investigating single category assignment (feature selection method, learning algorithms, categorization framework and performance evaluation functions are all based on the assumption that a document can be assigned to one category at the most). We also separate the training set and the testing set using the same split adopted by Lewis et al.

Classifier description and experimental setup

Algorithm 1 Proposed label evaluation method for LCN.

$\mathcal{T}r$, $\mathcal{T}v$ and $\mathcal{T}e$ are training, validation and testing sets, respectively. Θ is the classifier algorithm, which can be applied to any node of the hierarchy.

Training

For each node of the hierarchy c_i do:

1. Define the $T_{c_i}^+$ and $T_{c_i}^-$ from the $\mathcal{T}r$ set according to the policies given in Section 2.
2. Train a classifier $\hat{c}_i = \Theta(T_{c_i}^+, T_{c_i}^-)$

Validation

1. For $d \in \mathcal{T}v$ apply d to the all node classifiers
2. Calculate the recognition rate of each node on $\mathcal{T}v$ and use it as the weight factor for the node
3. For each d , calculate the reliability as follows:

$$Reliability(d) = \sum_{level=1}^L w(\hat{c}) \cdot CS(\hat{c}) \quad (2)$$

where $w(\hat{c})$ and $CS(\hat{c})$ are the weight factors and confidence score of the selected node at each level of the hierarchy.

4. Plot histogram of the mislabeled vs. truly labeled reliability score for $\mathcal{T}v$ set, specify the Equal Error Rate (EER) where the false acceptance (FA) and false rejection (FR) are equal, and regard it as the threshold τ

Testing

With $d \in \mathcal{T}e$:

1. Apply d to the classifiers in top-down manner, starting from the root downward to reach the leaf
2. Calculate the reliability score using formula 2 for the candidate route.
3. If $Reliability(d) > \tau$ then accept the assigned label otherwise, reject it or follow another classification strategy.
4. Calculate the boosted accuracy as follows:

$$Accuracy = \frac{\text{number of truly labeled samples}}{\text{number of accepted samples}} \quad (3)$$

Table 1. The main characteristics of the Reuter’s RCV1-v2 datasets.

problem	train	test	total samples	classes	levels	class/L2	class/L3	class/L4
topics	23,149	781,265	804,414	104	4	4	99	1
industry	23,149	781,265	804,414	365	4	10	354	1
regions	23,149	781,265	804,414	366	4	7	350	9

In TC applications the computational efficiency is crucial due to the very large number of features, classes, and samples size. Therefore, the issue of concerning the design of simple and fast classification systems is important. There are many research works in the literature using a variety of classifiers such as k-nearest neighbors (kNN), SVM, artificial neural networks, bayesian, and Rocchio's [14]. However, in practice most of them are not applicable since in real-world applications (e.g., search engines, contextual advertising, recommender systems) the real-time requirement has great importance. Among them, the Rocchio classification algorithm is extremely simple and straightforward while showing competitive performance on text categorization problems. Moreover, it does not require to store large amounts of training data. It calculates the prototype vector or centroid vector (C_i) for class node (c_i):

$$C_i = \frac{1}{|c_i|} \sum_{d \in c_i} d \quad (4)$$

where $|A|$ denotes the cardinality of set A and d is the training document.

In the testing step, we calculate the similarity of one document d to each centroid by the innerproduct measure,

$$S(d, C_i) = d \cdot C_i \quad (5)$$

This similarity can be regarded as the *posterior probability* of the node classifier and used for final decision making.

Moreover, to avoid the class-membership inconsistency problems, the node with max similarity have been selected at each level of the hierarchy. At the next level, the text sample have been applied only to the children of the selected node and so on till it reaches to the leaf. Therefore its also mandatory leaf-node prediction approach.

Performance results

In this subsection, we first show the performance of the proposed method in discriminating reliable vs. unreliable samples and then, rejecting the samples with the *reliability score* lower and including the samples whose *reliability score* is higher than the given threshold. False rejections (FR) occur when the label is *truly* assigned by the classifier while the reliability score is low or when the assigned label is false while high reliability is given to the sample in hand. Experimental results are reported in Table 2. As clearly shown, the proposed method rejects the samples falsely predicted by the classifier (TR), while the number of FR is very low when averaging on a large number of test samples. It is clear that the number of FR and TR are directly related to the selected route evaluation threshold. In particular, higher thresholds reduce TR while increasing the FR. Hence, in applications with high cost for mislabeling, the proposed strategy can reduce the overall cost with the drawback of rejecting more truly-labeled samples.

For the sake of comparison, different widely acknowledged standard text categorization algorithms have been run and evaluated on the selected

Table 2. The results obtained by the proposed method on the Reuter’s RCV1-v2 datasets.

problem	rejected samples	TR	FR	accuracy boost
topics	740	652	88	8.2
industry	602	580	22	6.5
regions	794	598	196	7.8

datasets. From these methods, the big-bang global method and flat are non-hierarchical while the LCP, LCL and LCN are hierarchical classification methods. To assess all the cited methods, a centroid-based classifier with the same parameters has been used. The results of this comparison is reported in Table 3, which clearly shows that the proposed method boosts the accuracy of the standard LCN method while outperforming both hierarchical and non-hierarchical methods.

Table 3. The proposed method outperforms the existing standard text categorization methods on the Reuter’s RCV1-v2 datasets. (Recognition rate in percentage)

problem	big-bang	flat	LCP	LCL	LCN	proposed method
topics	40.5	41.2	38.4	38.9	40.1	47.5
industry	44.3	43.0	42.1	41.3	42.7	47.4
regions	44.0	44.5	42.5	42.8	43.0	48.9

5 Conclusions and Future Work

A route confidence evaluation method is proposed for reliable HTC. The main strength of the proposed method concerns the integration of a prediction mechanism able to identify and deal with the samples which would be wrongly labelled by the classifier. This clearly results in a boost in accuracy of the LCN method by simply rejecting a low percentage of the test samples.

Experimental results on the Reuters RCV1-v2 datasets show a significant improvement in the recognition rate, compared to the standard LCN method. Furthermore, a comparison with results obtained by running other TC methods emphasizes an overall superior performance of the proposed method.

Acknowledgments. Camelia Chira acknowledges the support of Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

1. Koller D, Sahami M (1997) Hierarchically classifying documents using very few words. In: Proc. of the 14th Int. Conf. on Machine Learning, pp 170–178
2. Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5, 361–397
3. Yu B., Xu Z.-b., A comparative study for content-based dynamic spam classification using four machine learning algorithms (2008) *Knowledge-Based Systems*, 21 (4), pp. 355–362
4. S. Kiritchenko, S. Matwin, Email classification with co-training, *Intelligent Information Systems Conference*, pp. 523–533, 2004
5. X. Qi and B. D. Davison., Web Page Classification: Features and Algorithms. *ACM Computing Surveys*, 41(2), 2009
6. C. N. Silla Jr. and A. A. Freitas, A Survey of Hierarchical Classification Across Different Application Domains, *Data Mining and Knowledge Discovery*, vol. 20, no. 1, 2010
7. A. Kosmopoulos, E. Gaussier, G. Paliouras, S. Aseervatham, The ECIR 2010 Large Scale Hierarchical Classification, Workshop report, 2010
8. Eisner R, Poulin B, Szafron D, Lu P, Greiner R (2005) Improving protein function prediction using the hierarchical structure of the gene ontology. In: Proc. of the IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology, pp 1–10
9. Fagni T, Sebastiani F (2007) On the selection of negative examples for hierarchical text categorization. In: Proc. of the 3rd Language Technology Conference, pp 24–28
10. Ceci M, Malerba D (2007) Classifying web documents in a hierarchy of categories: A comprehensive study. *Journal of Intelligent Information Systems* 28(1):1–41
11. Wu F, Zhang J, Honavar V (2005) Learning classifiers using hierarchically structured class taxonomies. In: Proc. of the Symp. on Abstraction, Reformulation, and Approximation, Springer, 313–320, vol 3607
12. Dumais ST, Chen H (2000) Hierarchical classification of Web content. In Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval, pp 256–263
13. DeCoro C, Barutcuoglu Z, Fiebrink R (2007) Bayesian aggregation for hierarchical genre classification. In: Proc. of the 8th Int. Conf. on Music Information Retrieval, pp 77–80
14. F. Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys*, Volume 34 Issue 1, 2002.